課題解説

HRY

2019 年 5 月 22 日

では、R ではじめるデータサイエンスにおける指定問題 2.と 3.(p.67)の解説をします。 まず、準備段階として GitHub の Issue 述べた `install.packages(c("tidyverse", "nycflights13"))`を Rconsole で実行し、外部パッケージをインストール してください。

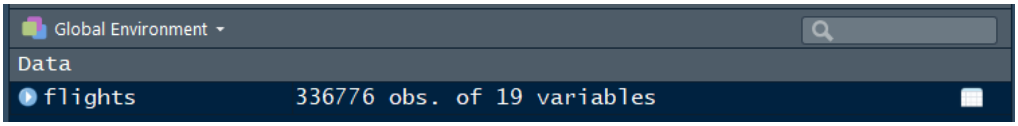
データ(flights)の確認

とりあえず、データセットの確認を行ってみましょう。

```
# パッケージの読み込み
library(tidyverse)

## -- Attaching packages ---------------------------------------------
------------------------------------------------------- tidyverse 1.2.
1 --

## √ ggplot2 3.0.0     √ purrr   0.2.5
## √ tibble  1.4.2     √ dplyr   0.7.7
## √ tidyr   0.8.1     √ stringr 1.3.1
## √ readr   1.1.1     √ forcats 0.3.0

## -- Conflicts ------------------------------------------------------
------------------------------------------------- tidyverse_conflicts
() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(nycflights13)

# Rstudio のGlobal Environment にflights オブジェクトを作成
flights <- flights
# 上から5 行までのデータの表示
head(flights)

## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
```

```
## 1  2013     1     1       517            515         2      830
## 2  2013     1     1       533            529         4      850
## 3  2013     1     1       542            540         2      923
## 4  2013     1     1       544            545        -1     1004
## 5  2013     1     1       554            600        -6      812
## 6  2013     1     1       554            558        -4      740
## # ... with 12 more variables: sched_arr_time <int>, arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <ch
r>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

`flights <- flights` というコードを Rstudio の Console で実行することによって、Rstudio の Global Environment にデータの行数(obs.)と列数(variables)が表示されます。



また、`View(flights)`を同様に実行することで Rstudio 上で文字通りデータを下図のように眺めることができます。

	year	month	day	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_delay	carrier	flight	tailnum	origin	dest	air_time	distance	hour	minute	time_hour
1	2013	1	1	517	515	2	830	819	11	UA	1545	N14228	EWR	IAH	227	1400	5	15	2013-01-01 05:00:00
2	2013	1	1	533	529	4	850	830	20	UA	1714	N24211	LGA	IAH	227	1416	5	29	2013-01-01 05:00:00
3	2013	1	1	542	540	2	923	850	33	AA	1141	N619AA	JFK	MIA	160	1089	5	40	2013-01-01 05:00:00
4	2013	1	1	544	545	-1	1004	1022	-18	B6	725	N804JB	JFK	BQN	183	1576	5	45	2013-01-01 05:00:00
5	2013	1	1	554	600	-6	812	837	-25	DL	461	N668DN	LGA	ATL	116	762	6	0	2013-01-01 06:00:00
6	2013	1	1	554	558	-4	740	728	12	UA	1696	N39463	EWR	ORD	150	719	5	58	2013-01-01 05:00:00
7	2013	1	1	555	600	-5	913	854	19	B6	507	N516JB	EWR	FLL	158	1065	6	0	2013-01-01 06:00:00
8	2013	1	1	557	600	-3	709	723	-14	EV	5708	N829AS	LGA	IAD	53	229	6	0	2013-01-01 06:00:00
9	2013	1	1	557	600	-3	838	846	-8	B6	79	N593JB	JFK	MCO	140	944	6	0	2013-01-01 06:00:00
10	2013	1	1	558	600	-2	753	745	8	AA	301	N3ALAA	LGA	ORD	138	733	6	0	2013-01-01 06:00:00
11	2013	1	1	558	600	-2	849	851	-2	B6	49	N793JB	JFK	PBI	149	1028	6	0	2013-01-01 06:00:00
12	2013	1	1	558	600	-2	853	856	-3	B6	71	N657JB	JFK	TPA	158	1005	6	0	2013-01-01 06:00:00
13	2013	1	1	558	600	-2	924	917	7	UA	194	N29129	JFK	LAX	345	2475	6	0	2013-01-01 06:00:00
14	2013	1	1	558	600	-2	923	937	-14	UA	1124	N53441	EWR	SFO	361	2565	6	0	2013-01-01 06:00:00
15	2013	1	1	559	600	-1	941	910	31	AA	707	N3DUAA	LGA	DFW	257	1389	6	0	2013-01-01 06:00:00
16	2013	1	1	559	559	0	702	706	-4	B6	1806	N708JB	JFK	BOS	44	187	5	59	2013-01-01 05:00:00
17	2013	1	1	559	600	-1	854	902	-8	UA	1187	N76515	EWR	LAS	337	2227	6	0	2013-01-01 06:00:00
18	2013	1	1	600	600	0	851	858	-7	B6	371	N595JB	LGA	FLL	152	1076	6	0	2013-01-01 06:00:00
19	2013	1	1	600	600	0	837	825	12	MQ	4650	N542MQ	LGA	ATL	134	762	6	0	2013-01-01 06:00:00
20	2013	1	1	601	600	1	844	850	-6	B6	343	N644JB	EWR	PBI	147	1023	6	0	2013-01-01 06:00:00
21	2013	1	1	602	610	-8	812	820	-8	DL	1919	N971DL	LGA	MSP	170	1020	6	10	2013-01-01 06:00:00
22	2013	1	1	602	605	-3	821	805	16	MQ	4401	N730MQ	LGA	DTW	105	502	6	5	2013-01-01 06:00:00
23	2013	1	1	606	610	-4	858	910	-12	AA	1895	N633AA	EWR	MIA	152	1085	6	10	2013-01-01 06:00:00
24	2013	1	1	606	610	-4	837	845	-8	DL	1743	N3739P	JFK	ATL	128	760	6	10	2013-01-01 06:00:00
25	2013	1	1	607	607	0	858	915	-17	UA	1077	N53442	EWR	MIA	157	1085	6	7	2013-01-01 06:00:00
26	2013	1	1	608	600	8	807	735	32	MQ	3768	N9EAMQ	EWR	ORD	139	719	6	0	2013-01-01 06:00:00
27	2013	1	1	611	600	11	945	931	14	UA	303	N532UA	JFK	SFO	366	2586	6	0	2013-01-01 06:00:00
28	2013	1	1	613	610	3	925	921	4	B6	135	N635JB	JFK	RSW	175	1074	6	10	2013-01-01 06:00:00

データの系列（変数）の確認については、`?flights` を実行することで確認できます。

flights {nycflights13}

Flights data

Description

On-time data for all flights that departed NYC (i.e. JFK, LGA or EWR) in 2013.

Usage

```
flights
```

Format

Data frame with columns

year,month,day

Date of departure

dep_time,arr_time

Actual departure and arrival times (format HHMM or HMM), local tz.

sched_dep_time,sched_arr_time

Scheduled departure and arrival times (format HHMM or HMM), local tz.

dep_delay,arr_delay

Departure and arrival delays, in minutes. Negative times represent early departures/arrivals.

hour,minute

Time of scheduled departure broken into hour and minutes.

carrier

Two letter carrier abbreviation. See `airlines()` to get name

tailnum

Plane tail number

- year,month,day：文字通りの意味で出発日に関する系列(Date of departure)
- dep_time,arr_time：実際の出発時刻と到着時刻の系列(Actual departure and arrival times (format HHMM or HMM), local tz.)
- sched_dep_time,sched_arr_time：予定出発時刻と予定到着時刻の系列(Scheduled departure and arrival times (format HHMM or HMM), local tz.)
- dep_delay,arr_delay：出発と到着の分単位の遅れの系列。負値は早い出発及び到着をあらわす(Departure and arrival delays, in minutes. Negative times represent early departures/arrivals.)
- hour,minute：出発予定時刻を時間と分に分割した系列(Time of scheduled departure broken into hour and minutes.)

- carrier：2 文字の業者の略語の系列(Two letter carrier abbreviation. See `airlines()` to get name)
- tailnum：飛行機の末尾番号の系列(Plane tail number)
- flight：フライト番号の系列(Flight number)
- origin,dest：出発地と目的地の系列(Origin and destination. See `airports()` for additional metadata.)
- air_time：飛行中に費やした時間（単位：分）系列(Amount of time spent in the air, in minutes)
- distance：空港間の距離（単位：マイル）系列(Distance between airports, in miles)
- time_hour：POSIXct 型のフライト予定日時系列。出発地とともに、フライトデータを気象データに結合するために使用される(Scheduled date and hour of the flight as a POSIXct date. Along with origin, can be used to join flights data to weather data.)

2.どの飛行機(tailnum)が定時離着陸記録に関して最悪か。

それでは、本題に移りましょう。方針としては、飛行機ごとに遅れまたは予定の早まりを集計すればよいということです。定刻きっかりに進行することが望まれると考えられるため、数値は絶対値として扱うこととします。

```
# tailnum のデータ数
data_tailnum <- flights %>%
  filter(!is.na(tailnum)) %>%
  group_by(tailnum) %>%
  summarise(N = n()) %>%
  arrange(desc(N))

# 飛行機の最大・最小数
max(data_tailnum$N)

## [1] 575

min(data_tailnum$N)

## [1] 1

# 遅れに関するデータオブジェクト data_delay
data_delay <- flights %>%
  filter(!is.na(dep_delay), !is.na(arr_delay)) %>%
  group_by(tailnum) %>%
  mutate(delay_score = abs(dep_delay) + abs(arr_delay)) %>%
  select(tailnum, delay_score) %>%
```

```
  summarise(score = sum(delay_score), N = n()) %>%
  mutate(mean_score = score / N) %>%
  arrange(desc(mean_score))

# deta_delay の確認
head(data_delay)
```

```
## # A tibble: 6 x 4
##   tailnum score     N mean_score
##   <chr>   <dbl> <int>      <dbl>
## 1 N844MH    617     1        617
## 2 N911DA    562     1        562
## 3 N922EV    550     1        550
## 4 N587NW    536     1        536
## 5 N851NW    452     1        452
## 6 N654UA    412     1        412
```

```
tail(data_delay)
```

```
## # A tibble: 6 x 4
##   tailnum score     N mean_score
##   <chr>   <dbl> <int>      <dbl>
## 1 N902DA      4     1        4
## 2 N693SW      7     2        3.5
## 3 N27901      3     1        3
## 4 N626AW      3     1        3
## 5 N456UW      2     1        2
## 6 N7BVAA      1     1        1
```

では、コードについて解説します。data_tailnum では飛行機の便数に関するデータオブジェクトを作成しています。2 行目の filter 関数(p.39)で tailnum が欠損値である行を行まるごと除き（リストワイズ）、tailnum を group_by 関数(p.63)によってグルーピングし、summarize 関数(p.55)で飛行機名毎の総便数 N を求めています。N の最大と最小を確認すると、最大は 575 便、最小は 1 便の運航となっており、飛行機によって N に大きな差があることが分かります。

この結果を受け、1 便当たりの遅れ時間 mean_score を求め、その値が最も大きいものを最悪な飛行機と呼ぶことにします w

data_delay データオブジェクトにおいて、mean_score を求めることとします。まず、filter(!is.na(dep_delay), !is.na(arr_delay))でキャンセル便をリストワイズします。そして、続く行の mutate 関数(p.48)で定刻との差の絶対値を足し合わせることで delay_score を求め、5 行目で select 関数(p.45)を用いて、必要とする系列(tailnum,delay_score)を選択しています。そして、6 行目で tailnum 毎の定刻との差の総和と総便数を求め、mutate で mean_score 系列を追加し、最後に

arrange(desc)を用いることで、mean_score を基準とする降順に並べ替えています。

結果的に、data_delay の先端部を確認することで、*N844MH* という飛行機が最悪であると分かります。1 便だけですが、約 10 時間分の定刻との差はいただけませんね。逆に最高？な飛行機は、こちらも 1 便だけですが *N7BVAA* という結果となりました。

それと、定刻との差が 0 である飛行機名は存在しませんでした。飛行機運航は定刻通りに進むことはとても難しいようです。私も初めての飛行機旅行として、中部国際-新千歳行きの便を利用したとき、出発が 2 時間ほど遅れるというアクシデント遭遇した経験があります。

3. 遅延をできるだけ避けたいとすれば、どの時間に飛行するとよいか。

次の問題に移ります。まず、私の考えでは問題文で「遅延を避けたい」とされているので、遅延スコア値（dep_delay と arr_delay）の和 delay_value が負であるものはリストワイズしてしまえばいいと思います。なぜ和による遅延スコア値で判断するべきかというと、例えば、出発は遅れたが到着は早まって結果的に和が 0 以下となる（先の遅れを取り戻した）ケース（逆もまた然り）が想定されるためです。

そして、先程の問題と同様に出発予定時間 hour でグルーピングし、集計するという方針とします。

```
# dep_delay または arr_delay が負のものをリストワイズ
data_avoid_delay <- flights %>%
  filter(!is.na(dep_delay), !is.na(arr_delay)) %>%
  mutate(delay_value = dep_delay + arr_delay) %>%
  filter(delay_value >= 0) %>%
  group_by(hour) %>%
  select(hour, delay_value) %>%
  summarise(sum_delay = sum(delay_value)) %>%
  arrange(sum_delay)

# 最適な時間
BestWorst_hours <- c(data_avoid_delay$hour[1], data_avoid_delay$hour[nrow
(data_avoid_delay)])
print(paste0("遅延をできるだけ避けるには、", BestWorst_hours[1], "時発の便を
選択するとよいでしょう"))
```

```
## [1] "遅延をできるだけ避けるには、5 時発の便を選択するとよいでしょう"
```

```
print(paste0("一方、遅延を最も避けにくいのは、", BestWorst_hours[2], "時発の便です"))
```

```
## [1] "一方、遅延を最も避けにくいのは、17 時発の便です"
```

コードについては、先の問題と比べて真新しいものは存在しないので割愛させていただきます。

結果として、5 時発の便を選ぶのがよいという結果となりました。各出発時刻ごとの結果を下に示します。パッと見た感じ、午前中に出発する便の方が遅延が少なそうに思われます。

	hour	sum_delay
1	5	16260
2	23	34813
3	22	109378
4	7	253489
5	6	280319
6	11	329367
7	10	330191
8	9	333201
9	12	413945
10	8	432967
11	21	541258
12	13	548162
13	14	699799
14	20	818043
15	15	884204
16	16	921523
17	18	976541
18	19	1060918
19	17	1110715

もし内容について御質問やご指摘がありましたら、こちらの GitHub の Issue で対応しようと思います。